Intrusion Detection Project Report

Farhat Lamia Barsha

# 1 Abstract

This report presents an analysis of my efforts in implementing decision-tree learning algorithm and assesses the efficacy of decision-tree learners within a broader scope. The utilization of decision trees is prevalent in the fields of machine learning and data analysis, hence emphasizing the significance of understanding their development and efficacy.

# 2 Introduction

The decision tree algorithm is a machine learning technique that has the capability to identify and classify instances of intrusion. The functioning of this system involves the creation of a decision tree, whereby each node within the tree corresponds to a decision and each leaf node corresponds to a prediction. The decision-making process relies on the characteristics or features of the data, whereas the predictions are made based on the target variable. In order to identify instances of intrusion through the utilization of a decision tree, the algorithm initially proceeds by training the tree on a dataset that comprises benign and malicious traffic. Subsequently, the algorithm utilizes the trained tree to make predictions regarding the classification of new data points as either intrusions or non-intrusions. Through the process of learning, the algorithm discerns how to divide the data by utilizing various features. One of the key benefits of decision trees is their interpretability, which facilitates the comprehension and examination of incidents for security analysts. Furthermore, these models exhibit high efficiency in real-time classification tasks, demonstrate adaptability to imbalanced datasets commonly seen in intrusion detection scenarios, and are particularly well-suited for networks of low to moderately big proportions.

# 3 Development of Decision Tree

My implementation of the decision tree algorithm for intrusion detection is divided into several parts. Here I will discuss every parts in details:

- **Part 1 - Dataset Preparation:** The process starts with loading a training dataset in ARFF format that has names of attributes. To read and work with this data, I use Pandas and SciPy. The attribute names are taken from the DataFrame and a translation dictionary is made to change the column names in the training and test datasets into names that are easier to understand. Redundant columns are removed from both the training and test datasets. This part makes sure that the data is organized correctly and that the attribute names are mapped correctly so that it can be used to build and test a decision tree-based intrusion detection model.

- **Part 2 - Algorithm Implementation:** In this section, the decision_tree_learning function is introduced, which is responsible for constructing a decision tree in a recursive manner. The implementation incorporates multiple auxiliary functions, including all_same_classification, majority_value, argmax_importance, and calculate_entropy.
  all_same_classification checks if all examples in the dataset have the same classification. majority_value calculates and returns the majority class label in a set of examples. argmax_importance selects the best attribute by calculating information gain based on entropy. calculate_entropy computes the entropy of the dataset, which is used to determine information gain. Additionally, a function named "pre-order_traverse" is defined to traverse the decision tree in a pre-order manner and display its nodes for visualization.

- **Part 3 - Generating Decision Tree:** This section starts by presenting a definition of a set of attributes, which encompasses the characteristics that will be utilized by the decision tree for the purpose of classification. The produced tree is subsequently employed to make predictions on both the training and test datasets using the predict function. The model's accuracy in classifying network traffic is determined by evaluating and storing the predicted labels.

# 4 Result and Analysis

I got 99.99% training accuracy and 65.76% testing accuracy. Here is a snapshot of my result:

```
Training Accuracy: 99.98729886562994 %
Test Accuracy: 65.76472675656494 %
```

This result indicates that the decision tree model exhibits outstanding performance on the training data, with an accuracy rate of around 99.99%. A training accuracy that is exceptionally high may indicate the presence of overfitting, wherein the model has achieved a near-perfect fit to the training data, but its ability to generalize to unseen data may be compromised.
In contrast, the test accuracy exhibits a notable decrease, approximately reaching 65.76%. The observed phenomenon is indicative of overfitting, wherein the model's ability to generalize to novel, unknown data is compromised. Addressing this issue is crucial in order to enhance the performance of the model when applied to real-world data.

# 5 Conclusion

The implementation of a decision tree for intrusion detection is a valuable approach for distinguishing between normal and intrusive network traffic. The provided implementation demonstrates the process of constructing a decision tree using training data. This decision tree is subsequently employed to categorize both the training and test datasets. The evaluation of the data is conducted based on accuracy, which serves as a metric to assess the efficacy of the model in detecting network intrusions. Nevertheless, it is crucial to recognize that although decision trees possess interpretability and efficiency, they might not fully encompass the intricacy of all intrusion patterns, particularly in the face of advancing and sophisticated threats. Hence, it is recommended to augment decision tree-based methodologies with more sophisticated techniques in order to achieve a comprehensive intrusion detection system.